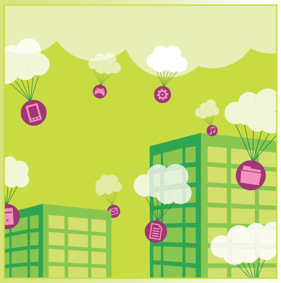

Fog Computing for the Internet of Things: Security and Privacy Issues

Fog computing has been introduced as a technology to bridge the gap between remote data centers and Internet of Things (IoT) devices. Enabling a wide range of benefits, including enhanced security, decreased bandwidth, and reduced latency, fog is an appropriate paradigm for many IoT services. Nevertheless, fog devices (located at the edge of the Internet) obviously face many security and privacy threats. Here, the authors discuss the security and privacy issues in IoT environments and propose a mechanism that employs fog to improve the distribution of certificate revocation information among IoT devices for security enhancement.

Whereas the cloud is up there in the sky somewhere ... the fog is close to the ground, right where things are getting done.[1]

Arwa Alrawais, Abdulrahman Alhothaily, Chunqiang Hu, and Xiuzhen Cheng
George Washington University

Fog computing is a new paradigm of distributed computing to extend the cloud to the network edge and provide efficient data access, computation, networking, and storage. Fog computing enables a new breed of services at the edge to deliver a wide variety of applications for Internet of Things (IoT) devices. It also supports mobility, location awareness, heterogeneity, large scalability, low latency, and geodistribution. Generally speaking, the goals of the fog computing paradigm are to reduce the data volume and traffic to cloud servers, decrease latency, and improve quality of service (QoS). The growing ubiquity of connected devices and sensors in smart grids, health care systems, wireless sensor networks, and smart homes forms a collective computing network, or the IoT. According to Gartner, 2.6 million IoT devices will be used in 2016, which is an increase of about 30 percent from 2015.[2] The increasing number of IoT devices will accordingly inflate the amount of data generated. IoT devices have inherently challenging characteristics such as low computation power, bandwidth, battery, and storage. In return, these characteristics affect the user experience and QoS. Typical IoT applications have a real-time requirement that can't be fulfilled with traditional cloud computing. We claim that fog computing offers an ideal solution to tackle these challenges and provide services with elastic resources at a low cost. Fog computing can serve as a potential solution due to its distinct characteristics such as location-awareness. Several works[3,4] have addressed the concept of applying fog computing to IoT applications such as vehicle ad hoc networks and health monitoring systems. This research discussed how fog computing can improve multiple aspects in IoT applications.

Fog is considered as a nontrivial extension of the cloud; therefore it's inevitable that some security and privacy challenges will continue to persist. While some existing solutions in the context of cloud computing could address many security and privacy issues in fog computing, fog could introduce new security and privacy challenges due to its distinct characteristics such as mobility support. These challenges might impact the adaptation of fog computing into the IoT. On the other hand, fog computing could offer an ideal platform to address many security and privacy issues in the IoT. The balanced mixture of computational power, connectivity, and span of control lets fog support IoT applications and serve various security purposes. Fog nodes can be represented as proxies that provide cryptographic computations, while IoT devices and sensors underneath lack the necessary resources to do so. Therefore, fog computing could provide not only additional computational resources, but also an unprecedented level of security that will help minimize attacks in IoT environments.

The research on the security and privacy issues of fog computing for the IoT is still in its early stage. In this article, we take a closer look at the security and privacy issues of fog computing in IoT environments. Moreover, as a case study, we propose a new scheme using fog to solve security issues in distributing certificate revocation information in IoT environments.

Security and Privacy Challenges in IoT

Although the IoT can play a central role in delivering a rich portfolio of services more effectively and efficiently to end users, it could impose security and privacy challenges. In the following, we summarize the major security and privacy challenges in IoT environments.

Authentication

Authentication is an essential requirement for the security of IoT devices. Unfortunately, many IoT devices don't have enough memory and CPU power to execute the cryptographic operations required for an authentication protocol. These resource-constrained devices can outsource expensive computations and storage to a fog device that will execute the authentication protocol. Yee Wei Law and colleagues[5] proposed a wide-area measurement system key management (WAKE) model for the smart grid. This model is based on public-key infrastructure (PKI) using multicast authentication for secure communications. While traditional PKI-based authentication could solve the problem, it wouldn't scale well for IoT systems.

Trust

Due to the nature of the IoT environment, which integrates various devices and sensors belonging to multiple actuators, the following question arises: To what degree can we trust the IoT devices? There's no efficient mechanism that can measure when and how to trust IoT devices. In the absence of a trust measurement, users of IoT services need to consider whether it's profitable to abstain from using certain IoT services. Therefore, cultivating the trust between IoT devices plays a central role in establishing secure environments to preserve the security and reliability of IoT services. Trust models based on reputation have been successfully deployed in many scenarios such as online social networks. Kai Hwang and colleagues proposed a new approach to improve the trust in clouds, which combines security-reinforced data centers, data access, and virtual clusters directed by reputation systems.[6] To design a trust model based on reputation in the IoT,[3] we need to tackle how to maintain the service reliability and prevent accidental failures, handle and identify misbehavior issues, identify malicious behavior correctly, and bootstrap building a trust model based on reputation in large-scale networks.

Rogue Node Detection

A malicious IoT node could pretend to be legitimate to exchange and collect the data generated by other IoT devices for malicious purposes. Liran Ma and colleagues proposed a hybrid framework that can detect the presence of rogue access points in Wi-Fi-based access networks.[7] Their approach protects the networks from rogue access points even if the adversaries use customized equipment. A rogue IoT node has the potential to misuse users' data or provide malicious data to neighboring nodes to disrupt their behaviors. Addressing this problem could be difficult in the IoT due to the complexity in trust management in various schemes. However, a trust measurement-based model could be applied to detect rogue nodes in IoT environments, which can provide limited security protection.

Privacy

The privacy leakage of user information in IoT environments, such as data, location, and usage, is attracting the attention of the research community. The resource-constrained IoT devices lack the ability to encrypt or decrypt generated data, which makes it vulnerable to an adversary. Another privacy issue is the location privacy that can be used to infer the IoT device's location. Several IoT applications are location-based services, especially mobile computing applications. An adversary can infer the IoT device's location based on the communication patterns. The last privacy issue is the protection of a user's usage pattern of some generated data by IoT devices, such as in the smart grid. For instance, the readings of smart meters can reveal many usage patterns of IoT clients, such as how many people live in the household, when they turn on the TV, or when they are at home. Many privacy-preserving schemes have been proposed in different IoT applications such as smart grids, healthcare systems, and vehicle ad hoc networks.[3,8] However, the resource-constrained IoT devices limit the techniques that can be used to deliver efficient and effective privacy-preserving schemes.

Access Control

Access control is a security technique to ensure that only authorized entities can access a certain resource, such as an IoT device, or the collected data. In the IoT, we need access control to make sure that only trusted parties can perform a given action such as accessing IoT device data, issuing a command to an IoT device, or updating IoT device software. The IoT introduces new challenges in access control because we're dealing with a huge number of "things" that have limited resources (that is, power and bandwidth). Besides, managing access to highly distributed data is by itself a significant challenge.

Intrusion Detection

Intrusion detection techniques detect misbehavior or malicious IoT devices and notify others in the network to take appropriate actions. Most of the existing techniques in the IoT target a few attacks with low efficiency. The nature of IoT environments makes it difficult to detect the insider and outsider attacks in such universal platforms. Additionally, the complicated design of intrusion detection techniques that meets the limited resources in the IoT is another challenging task. The key challenge is how to design and tune a detection system that can work in large-scale, widely geodistributed, and highly mobile environments.

Data Protection

The exponential volume of data generated by IoT is growing with the increasing number of devices. This data must be preserved not only at the communication level, but also at the processing level. Due to the resource limitations, it's difficult to process the data on IoT devices; hence, data usually is sent to the cloud for further processing and analyzing. At this point, the data integrity should be preserved during and after the processing stage. The lack capability of IoT devices to encrypt or decrypt makes computing the authenticity and integrity of the data a critical challenge.

Other Challenges

The aforementioned security and privacy issues in IoT environments are illustrative and not exhaustive. There are other security challenges such as key management, data aggregation, and verifiable computing. Nevertheless, the distinguished characteristics of fog computing can contribute to address the issues related to security and privacy in IoT environments. Furthermore, fog computing could be a part of the security solution to ensure that IoT services aren't vulnerable to the most common attacks in IoT, such as denial of service (DoS) attacks and malware-based attacks. Under DoS attack scenarios, for example, the wide distribution of fog nodes could help preserve the resiliency for IoT services.

In the following section, we argue that fog computing can help the IoT tackle several security and privacy issues. We elaborate the role of fog computing in the certificate revocation distribution by designing an efficient scheme that meets the special requirements of IoT devices.

Most current and proposed certificate revocation schemes have many limitations such as high bandwidth use and timeliness issues that could lead to serious security consequences. We propose a certificate revocation scheme that enhances the security and alleviates consuming the network bandwidth by using the fog device as a gateway in IoT environments to distribute the certificate revocation information.

Case Study: Resorting to Fog for Certificate Revocation in the IoT

To begin, we briefly summarize the most popular certificate revocation schemes and discuss their limitations in the IoT. Next, we propose a new certificate revocation scheme in IoT based on fog computing. Finally, we discuss and analyze the results of our scheme.

Certificate Revocation Schemes

The two predominant schemes for disseminating revocation information are certificate revocation list (CRL) and Online Certificate Status Protocol (OCSP).

A CRL is a file that maintains all revoked certificate serial numbers that should no longer be trusted. To trust a certificate, a client must download the CRL file and check whether the intended certificate is listed. Each certificate authority (CA) issues a CRL file and signs it to ensure its integrity. The CRL file is released by CAs periodically. Some CAs issue a CRL with a long window between two consecutive releases such as a week or a month. This delay could lead to security incidents such as accepting a revoked certificate as a valid one. An earlier real-world CRL study indicated that about 30 percent of the issued certificates were revoked within the first two days of the certificate's lifetime.[9] Therefore, a CA that releases its CRL file with a longer interval time could impose serious security risks. It's a best practice that all involved entities are notified whenever a certificate is revoked. Additionally, the cumulative number of the revoked certificates makes the CRL file size grow over time. This growth will incur an extra communication overhead every time the CRL file is transferred to a client. Consequently, the CRL file consumes the clients' resources such as storage and power to download and store the CRL file. Generally speaking, CRLs have been criticized for not being effective in distributing certificate revocation information.

OCSP is an online protocol used to obtain the digital certificate status. A client submits an OCSP request containing the certificate's serial number that needs to be checked by the OCSP responder server. In return, the OCSP responder checks its up-to-date revocation list and provides the client with a signed response indicating whether the certificate status is good, revoked, or unknown. As we can conclude, OCSP overcomes many limitations in CRLs. First, it doesn't require large storage at the client side because OCSP is an online protocol. Second, unlike the CRL, OCSP doesn't incur expensive communication overhead, because OCSP message size is small compared with CRL file size. Finally, the OCSP provides an immediate update about the certificate status, which means that many attacks are prevented. Nevertheless, OCSP checking potentially impairs clients' privacy and imposes a high latency overhead for each request. The round trip between the client and the OCSP responder server adds a significant amount of time to the verification process. A recent study showed that the median OCSP responder time is about 291 ms.[10]

The public key cryptosystem can maintain the security of communications in large-scale networks. However, new challenges are raised when it comes to secure IoT communications. One of these challenges is to distribute the certificate revocation information efficiently among a huge number of IoT devices. The certificate revocation schemes described previously have many issues, which make them unsuitable for IoT environments. In the CRL method, the characteristics of IoT make the CRL an inappropriate revocation scheme. A CRL file imposes a high burden of communication overhead and requires large storage on the client side. The size of a CRL file correlates with the number of the revoked certificates, and it grows over time. In other work, we analyzed the size of the CRL files and updated period time for the top 10 CAs in the TLS handshake for the Alexa top 1 million domains dataset.[11,12] The CRL files' size varies between 793 bytes and 5 Mbytes, and the updated period time is ranging from a few days to a year. Accordingly, a CRL wouldn't perform well in IoT environments due to the resource limitations of IoT devices. In comparison, OCSP is a real-time protocol, making it more suitable for IoT deployment. However, OCSP requires a round-trip time to check each certificate's

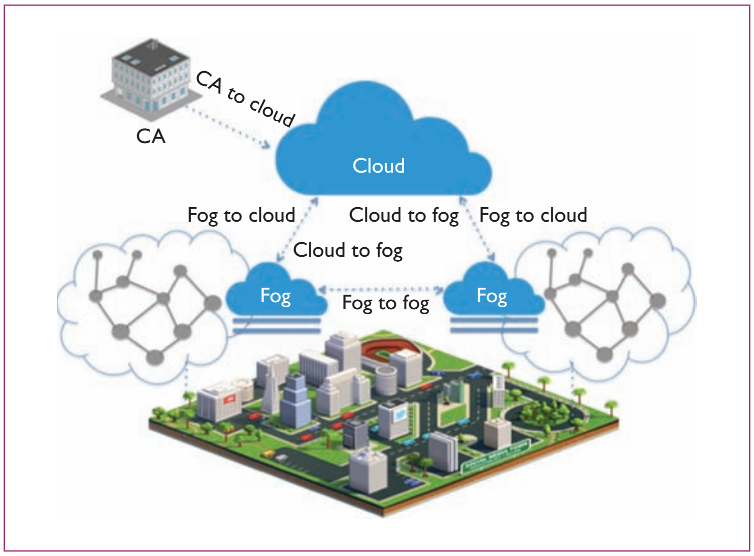

Figure 1. An overview of our scheme. The scheme consists of a certificate authority (CA), a back-end cloud, fog nodes, and Internet of Things (IoT) devices.

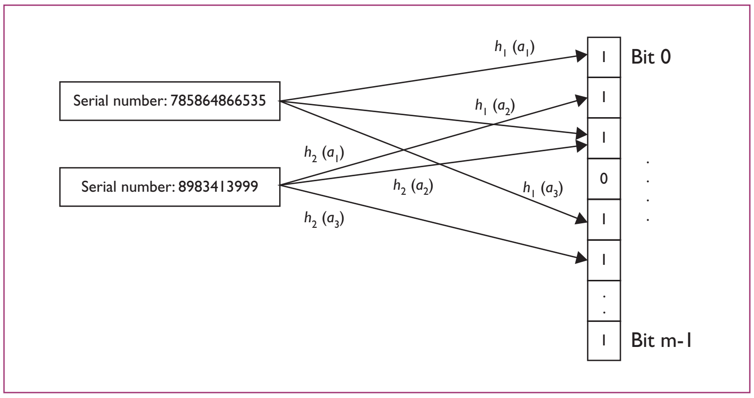

Figure 2. Example of a bloom filter with k = 3 independent hash functions. Although a false positive matching is possible in the bloom filter, a false negative is not.

status, and IoT devices can't afford to communicate each time with the OCSP server due to resource constraints.

Overview of Our Scheme

We describe our scheme and show how it uses the fog computing paradigm to improve the efficiency and effectiveness of certificate revocation distribution in IoT environments. Our scheme consists of a CA, a back-end cloud, fog nodes, and IoT devices (see Figure 1). Note that we assume that each fog node is responsible for serving a group of IoT devices that use digital certificates issued by a particular CA, and our scheme easily can be generalized to the case of one fog node responsible for certification revocation of multiple CAs.

In our scheme, we use a bloom filter to create a short list that can effectively reduce the revocation list size with acceptable overhead. A bloom filter is a space-efficient data structure to store a group of elements in a bit-vector (0, ..., $m - 1$) that can be used to check whether an element is a member of the group. Initially, an empty bloom filter vector is set to 0. To store certificate revocation information, we use the certificate's serial number, which is considered to be its unique identifier within a CA. The serial number should be hashed with k independent hash functions mapping it to a group of bit locations that should be set to 1. Figure 2 shows the essential idea of the bloom filter. Although a false positive matching is possible in the bloom filter, a false negative is not. We explain how to resolve this problem in our scheme in the following description.

Each CA creates and signs an updated CRL; then sends it to a cloud. The purpose of using the cloud in our scheme is to manage the received revocation lists from multiple CAs and forward them to the intended fog nodes, because each fog node serves a group of IoT devices that hold certificates belonging to a specific CA. Then, the fog node stores the list, and prepares a bloom filter that maps the revocation information. Also, the fog node signs the bloom filter to preserve its integrity and distributes the bloom filters to the associated IoT devices. After receiving the bloom filter, each IoT device should verify the fog's signature and store the vector.

When an IoT device needs to communicate with another device, it needs to verify the device's certificate status. If the certificate's identity isn't in the bloom filter, it means the certificate isn't revoked. However, if the identity is found in the bloom filter, there are two possible cases: first, the certificate is revoked; or second, the certificate isn't revoked and the bloom filter triggers a false positive result. To resolve this issue, the fog node acts as a gateway for each IoT group to check the certificate's status. If the certificate is found in the bloom filter, the device needs to contact the fog node to verify the certificate's status. The IoT device sends to the fog a packet containing the intended device's certificate serial number. Then, the fog checks the certificate's serial number against the stored list and replies with the certificate's status.

Results and Analysis

The proposed scheme provides an efficient distribution of the certificate revocation information. Our scheme is an up-to-date approach where all revoked certificates are sent immediately from the CAs to the cloud, then to the fog computing devices. Immediate updating of the revocation information ensures the security of the certificate validation process and eliminates the risk of accepting a revoked certificate. Another security feature is that the fog signs each bloom filter to provide proof of its reliability. Because it's infeasible to modify or fabricate the fog's signature, this signature represents an undeniable proof that the vector has originated from the fog. In our scheme, the advantage of using the bloom filter is that it can reduce the computation overhead on IoT devices because the bloom filters use an efficient hashing operation to verify the certificate status. Furthermore, it's well-known that the bloom filter is a space-efficient data structure that will preserve IoT device resources such as storage. However, the false positive probability is the price paid for the space and time efficiency of the bloom filter. Adjusting the number of hash functions k and the bloom filter size can affect the false positive probability. Other work provides more details about bloom filters.[13]

We now present a quantitative evaluation of our scheme, and compare it with the current revocation certificate schemes. Our main concern is the resource consumption on IoT devices such as storage and communication overhead.

Storage. Because the bloom filter size is directly related to the resource consumption, we start from the point of analyzing the bloom filter size in our scheme and compare it with other revocation approaches. We assume that each fog node has a group of IoT devices with n valid certificates, and the average number of revoked certificates per day is b. In our proposed scheme, the total size of the bloom filter can be computed as follows:

$$m = \frac{-b \cdot ln(p)}{(ln(2)^2)}$$

where m is the number of bits needed in the bloom filter, and p is the chosen probability of a false positive, which is 0.01 in our experiment.[13] To compute the CRL file size, we need to consider the certificate's serial number length, and the CA's signature length. A typical certificate's serial number length ranges from 15 to 20 bytes, and a CA's signature length is about 700 bytes. The total CRL file size is $b \cdot 20 + 700$ bytes. For OCSP, the protocol doesn't incur any storage overhead, because it's an online approach.

Communication overhead. From the communication aspects, the main contributor to the communication overhead is the packet size. The overhead in terms of bandwidth consumption can be computed using the packet size and connection speed. Obviously, the packet size is different for each revocation scheme. Following the previously mentioned assumption where the number of revoked certificates per day is b, we can calculate the daily packet size for each revocation scheme. In CRL, the file size is calculated to get $b \cdot 20 + 700$ bytes. Similarly, for OCSP, we need to consider the number of communications issued per day to compute the total number of OCSP packet sizes, because an OCSP query is issued to check the certificate status regarding whether it's revoked or not.

Let's assume that at least d of the IoT devices within each group need to communicate and exchange data every day, and $d \geq b$. Consequently, to get a precise evaluation, we only need to compute b OCSP request and b OCSP response packets in our evaluation. Our experiment shows that the average OCSP request packet size is 140 bytes, and the average OCSP response packet size is 152 bytes. The total packet size for an OCSP query is 292 bytes considering both the request and response packet sizes. Figure 3 illustrates the packet size per day for different revocation schemes when the number of revoked certificates b is 50, 30, or 10. We can see that our scheme requires a significantly smaller packet size than OCSP and CRL. To compute the communication overhead, we need to consider the average connection speed, which is assumed to be 10 Mbps. We can compute the bandwidth consumption for each revocation approach, as Figure 4 shows. Our scheme consumes less bandwidth, which is compatible with the packet size.

The proposed scheme is a perfect illustration on how the fog computing paradigm can enhance the security of IoT applications. Our scheme uses the fog device to act as a gateway for IoT devices to distribute the revocation

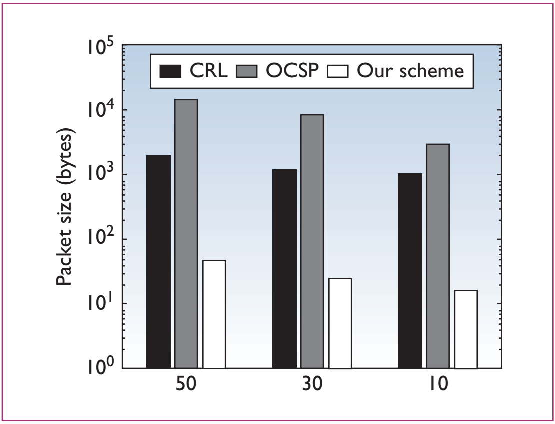

Figure 3. Comparison of the packet size in different certificate revocation schemes when b = 50, b = 30, and b = 10. CRL = certificate revocation list and OCSP = Online Certificate Status Protocol.

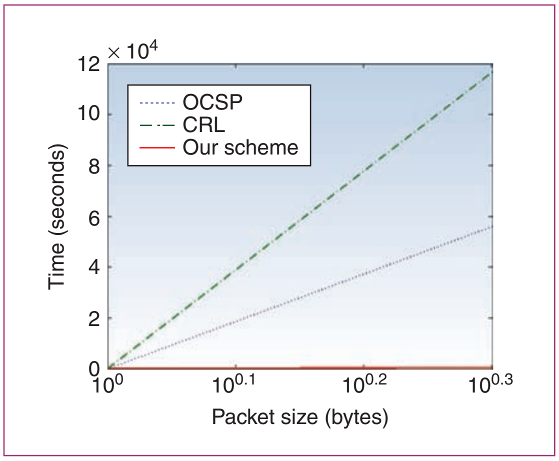

Figure 4. Comparison of the communication overhead in different certificate revocation schemes. Our scheme consumes less bandwidth, which is compatible with the packet size.

information in a timely, efficient, and secure manner. Next, we discuss potential fog-based solutions to improve the security and privacy issues in general IoT environments.

Open Research Issues: Fog Computing and Security and Privacy in IoT Environments

IoT devices must provide trustworthy information to the providers, clients, and other IoT devices. Establishing secure communications across IoT devices on a massive scale, of course, is a critical challenge. It's difficult to have secure channels on the grand scale of the IoT, but it's indisputable that fog computing is going to play a fundamental role in addressing security and privacy issues in IoT applications. Here, we summarize the major open research challenges for enhancing security and privacy issues in IoT using the fog computing paradigm.

Privacy

The fog computing paradigm could help preserve privacy in the IoT and protect users by minimizing the need to transmit sensitive data to the cloud for analysis. Using this new model would help in analyzing and processing the data at the network edge, close to the IoT devices that generate and act on that data. However, this approach introduces several challenges concerning data, location, and user privacy. In fog computing, privacy-preserving techniques could be applied between the fog and the cloud to preserve data privacy because both have sufficient storage and power. It's challenging, though, to run these techniques between the fog and IoT devices due to IoT resource limitations. One possible solution is a privacy-preserving technique based on homomorphic functions that could be deployed to maintain the transmitted data's privacy. Differential privacy is another technique to ensure nondisclosure of privacy in the dataset.[3] Nevertheless, the computational overhead of such approaches results in big concerns.

To preserve location privacy, one initial solution is to allow the IoT to distribute the data among several fog nodes. This solution would waste fog resources and increase the delay time. An identity obfuscation technique[8] can be used in fog computing for IoT devices such that fog nodes can't identify which IoT device is offloading the data. Another possible solution to protect user privacy would be designing an efficient privacy-preserving technique based on partitioning the data among fog devices.

Updating IoT Devices

Unfortunately, many IoT devices are still vulnerable, and remote software update capabilities need to be designed to handle security updates. Fog computing could be a crucial part of a solution that identifies vulnerabilities and tracks firmware updates in IoT devices. Vulnerable firmware can leave IoT devices open to attacks against which traditional security solutions (such as firewalls) might not be effective.

Updating billions of IoT devices is a cumbersome task, but the geodistribution characteristic of fog computing could help supply IoT devices with the necessary security updates to keep them secure.

Secure and Efficient Protocols

Many existing protocols such as time synchronization are based on wireless packet transmissions, and they aren't suitable for resource-constrained IoT devices. Wireless transmissions and security computations consume a significant fraction of the energy budget. The primary challenge is how to design efficient secure schemes in IoT without sacrificing the performance and consuming a high energy.

Authentication

Authentication in IoT has several challenges such as scalability and efficiency. Traditional authentication is inefficient, and there's a need for a secure, scalable, efficient, and user-friendly solution to cope with resource-constrained IoT devices. Facilitated by fog, a lightweight encryption algorithm can be applied between fog nodes and IoT devices to improve the efficiency of the authentication process. Furthermore, fog could create an opportunity for authentication in IoT devices, particularly wearable devices.[14,15]

Attack Detection

Fog computing provides new opportunities to detect unusual behavior and spot malicious attacks. In general, a detection system can be signature- or anomaly-based, in which the pattern can be compared or checked against existing possible patterns. Because the fog is an extension of the cloud at the network edge, it's possible to reuse developed detection systems of the cloud in the fog platform. Yue Shi and colleagues proposed a cloudlet mesh architecture that's based on a collaboration of the cloudlet members to observe and detect malware, malicious attacks, and other threats.[16] This type of collaborative intrusion-detection technique can be used between fog nodes to monitor IoT environments and their surroundings. Fog computing provides a new opportunity to design an efficient intrusion-detection solution on both the cloud and IoT device sides. Employing such a solution at fog nodes will add a layer of protection that monitors and detects any unusual behavior in IoT environments.

Location Verification

In harsh environments such as vehicles and railways, IoT devices can move in a fast and dynamic way, which complicates location verification. The main challenge is how to design a secure and precise location-verification scheme in harsh environments, and at the same time, the scheme should be suitable for resource-constrained IoT devices.

Access Control

With the help of fog computing, there's a need to design a new access control model that overcomes the limitations of IoT devices. A related work has proposed policy-based management to authorize users' access in fog computing.[17] Fog would facilitate the adoption of many standard access control models, such as an access control list or attribute-based access control in IoT environments. In a distributed architecture, we envision fog computing as an ideal candidate to grant access tokens to authorized parties who use them to perform a given action. Also, we can use the fog computing platform in a centralized architecture to authorize access and relay data between authorized parties and IoT devices.

We investigated and discussed security and privacy challenges of introducing fog computing in IoT environments. It's essential that greater attention should be focused on how to overcome these challenges, such as authentication in the context of fog computing in IoT applications. In our future work, we plan to explore and design security solutions to tackle the proposed challenges. Moreover, we will study new privacy issues such as location privacy.

Acknowledgments

Arwa Alrawais thanks the scholarship fund from the Ministry of Higher Education, Saudi Arabia, and the College of Computer Engineering and Sciences, Prince Sattam bin Abdulaziz University, Saudi Arabia. Abdulrahman Alhothaily thanks the scholarship fund from the Saudi Arabian Monetary Authority.

References

1. C. Mims, "Forget 'the Cloud'; 'The Fog' Is Tech's Future," *The Wall Street J.*, 18 May 2014; www.wsj.com/articles/SB10001424052702304908304579566662320279406.

2. Gartner, “Gartner Says 6.4 Billion Connected ‘Things’ Will Be in Use in 2016, Up 30 Percent from 2015,” press release, 10 Nov. 2015; www.gartner.com/newsroom/id/3165317.
3. S. Yi, Z. Qin, and Q. Li, “Security and Privacy Issues of Fog Computing: A Survey,” *Proc. Int’l Conf. Wireless Algorithms, Systems, and Applications*, 2015, pp. 685–695.
4. M. Al Faruque and K. Vatanparvar, “Energy Management-as-a-Service Over Fog Computing Platform,” *IEEE Internet of Things J.*, vol. 3, no. 2, 2012, pp. 161–169.
5. Y.W. Law et al., “Wake: Key Management Scheme for Wide-Area Measurement Systems in Smart Grid,” *IEEE Communications Mag.*, vol. 51, no. 1, 2014, pp. 34–41.
6. K. Hwang, S. Kulkareni, and Y. Hu, “Cloud Security with Virtualized Defense and Reputation-Based Trust Management,” *Proc. 8th IEEE Int’l Conf. Dependable, Autonomic, and Secure Computing* (DASC), 2009, pp. 717–722.
7. L. Ma, A.Y. Teymorian, and X. Cheng, “A Hybrid Rogue Access Point Protection Framework for Commodity Wi-Fi Networks,” *Proc. 27th IEEE Conf. Computer Comm.*, 2008; doi:10.1109/infocom.2008.178.
8. W. Wei, F. Xu, and Q. Li, “MobiShare: Flexible Privacy-Preserving Location Sharing in Mobile Online Social Networks,” *Proc. IEEE Conf. Computer Comm.*, 2012, pp. 2616–2620.
9. C. Ma, N. Hu, and Y. Li, “On the Release of CRLs in Public Key Infrastructure,” *Proc. 15th Usenix Security Symp.*, vol. 15, 2006, article no. 2.
10. E. Stark et al., “The Case for Prefetching and Prevalidating TLS Server Certificates,” *Proc. 19th Ann. Network & Distributed System Security Symp.*, 2012; www.internetsociety.org/sites/default/files/12_4.pdf.
11. A. Alrawais, A. Alhothaily, and X. Cheng, “X. 509 Check: A Tool to Check the Safety and Security of Digital Certificates,” *Proc. Int’l Conf. Identification, Information, and Knowledge in the Internet of Things*, 2015, pp. 130–133.
12. A. Alrawais et al., “Secureguard: A Certificate Validation System in Public Key Infrastructure,” to be published.
13. D. Miner and A. Shook, *MapReduce Design Patterns: Building Effective Algorithms and Analytics for Hadoop and Other Systems*, O’Reilly Media, 2012.
14. A. Alhothaily et al., “Towards More Secure Cardholder Verification in Payment Systems,” *Proc. Int’l Conf. Wireless Algorithms, Systems, and Applications*, 2014, pp. 356–367.
15. A. Alhothaily, A. Alrawais, and X. Cheng, “A Novel Verification Method for Payment Card Systems,” *J. Personal and Ubiquitous Computing*, vol. 19, no. 7, 2015, pp. 1145–1156.
16. Y. Shi, S. Abhilash, and K. Hwang, “Cloudlet Mesh for Securing Mobile Clouds from Intrusions and Network Attacks,” *Proc. 3rd IEEE Int’l Conf. Mobile Cloud Computing, Services, and Eng.*, 2015, pp. 109–118.
17. C. Dsouza, G.-J. Ahn, and M. Taguinod, “Policy-Driven Security Management for Fog Computing: Preliminary Framework and a Case Study,” *Proc. 15th Int’l Conf. Information Reuse and Integration*, 2014, pp. 16–23.

Arwa Alrawais is a PhD student in the Department of Computer Science at George Washington University. Her research interests include network security, wireless and mobile security, and algorithm design and analysis. Alrawais has an MS in computer science from George Washington University. She’s a student member of IEEE. Contact her at alrawais@gwmail.gwu.edu.

Abdulrahman Alhothaily is a PhD candidate in computer science at George Washington University. His research interests include payment security, fraud, wireless and mobile security, and security engineering. Alhothaily has an MS in computer science from George Washington University. He’s a recipient of the Best Student Award from George Mason University, and a PhD scholarship from the Saudi Arabian Monetary Authority. Contact him at hothaily@gwu.edu.

Chunqiang Hu is a postdoctoral research fellow at the Catholic University of America; he previously was a visiting scholar at George Washington University. His research interests include applied cryptography, big data security and privacy, privacy-preserving computations, wireless and mobile security, and algorithm design and analysis. Hu has PhDs in computer science and technology from Chongqing University, China, and in computer science from George Washington University. He’s a member of IEEE and ACM. Contact him at chu@gwu.edu.

Xiuzhen Cheng is a professor in the Department of Computer Science, George Washington University. Her research interests include privacy-aware computing, wireless and mobile security, and algorithm design and analysis. Cheng has a PhD in computer science from the University of Minnesota — Twin Cities. She received the National Science Foundation Career Award in 2004. She’s an IEEE fellow. Contact her at cheng@gwu.edu.